

VISUAL DATA MINING FOR QUANTIZED SPATIAL DATA

Amy Braverman and Brian Kahn

Key words: Massive data sets, cluster analysis, multivariate visualization.
COMPSTAT 2004 section: Applications.

Abstract: In previous papers we've shown how a well known data compression algorithm called Entropy-constrained Vector Quantization (ECVQ; Chou, Lookabaugh and Gray, 1989) can be modified to reduce the size and complexity of very large, satellite data sets. In this paper, we discuss how to visualize and understand the content of such reduced data sets. We developed a Java tool to facilitate this using simple multivariate visualization, and interactively performing further data reduction on user selected spatial subsets. This enables analysts to compare reduced representations of the data for different regions and varying spatial resolutions. The ultimate aim is to explain physically observed differences, trends, patterns and anomolies in the data.

1 Introduction

This work came about because of challenges posed by NASA's Earth Observing System (EOS). EOS is a long-term data collection program for studying climate change, its consequences for life on Earth, and effects of human activities on it. The centerpieces of EOS are three satellites, Terra, Aqua and Aura. Terra and Aqua are already in orbit, and Aura is due for launch in 2004. Each carries a suite of instruments that collect massive amounts of observational data; so massive that it is difficult to take full advantage of them. Different instruments have different sampling strategies, resolutions, file naming conventions, and collect data about different physical processes. The information is provided to users in files corresponding to individual spacecraft orbits or parts of orbits, each of which can be very large, and must be stitched together properly to provide a global or even a regional picture. To make these data more accessible, NASA produces global summary data sets called Level 3 data products.

Traditionally, Level 3 products are simple maps of mean quantities and standard deviations at coarse spatial resolution, by month. In Braverman (2002), we proposed methods for constructing nonparametric, multivariate distribution estimates to replace traditional maps. For instance, the Multiangle Imaging SpectroRadiometer (MISR) aboard Terra collects data about clouds. A key goal is to better understand the spatial distribution of clouds since they have great influence on Earth's energy budget. The information MISR collects includes three variables seen at high resolution: scene albedo,

height, and cloud presence indicator. Albedo is a measure of scene reflectivity measured roughly on a scale of zero to one. Scene height is measured in meters above the Earth's surface ellipsoid. The cloud indicator is a binary variable taking value one if the scene is cloudy, and zero otherwise. To summarize this information traditional Level 3 products are created by partitioning one month's data into spatial subsets corresponding to one degree latitude–longitude grid cells. Six maps are then produced: mean and standard deviation of albedo, mean and standard deviation of height, and mean and standard deviation of cloud indicator.

The Level 3 product we proposed regards each triplet of albedo, height and cloud indicator as a three-element vector, and uses ECVQ to cluster data each grid cell. We report a set of cluster representatives, the number of original data points belonging to each cluster, and within–cluster mean squared error, also called distortion. We call this a summary, or a compressed or quantized version of the grid cell's data. Figure 1 illustrates. For one grid cell it shows a three dimensional scatterplot of the original data in light gray. Positions of cluster representatives are shown by the embedded balls, and ball shading shows cluster population according to the color bar on the right. Two key features of the summary are that i) cluster representatives be

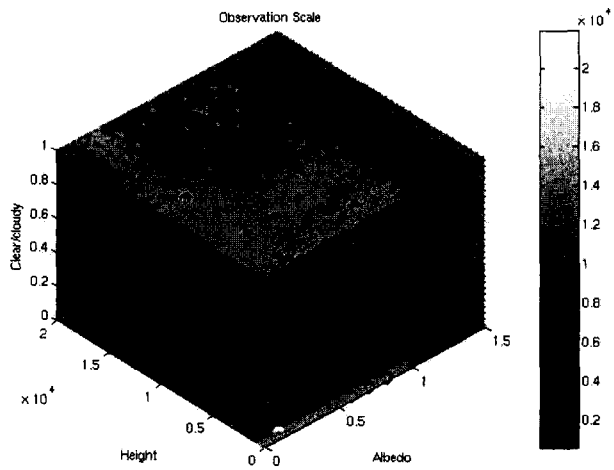

Figure 1: Three-dimensional scatterplot of MISR albedo, height and cloudiness data, in light gray, for a one degree grid cell in northern Oklahoma (southwest corner 38°N, 98°W) in March 2000. The embedded balls show the locations of cluster representatives. The ball colors show cluster populations using the gray-scale color bar on the right.

centroids of cluster members, and ii) data vectors must be assigned to clusters with the nearest (euclidian distance) representatives. This ensures that mean squared error between grid cell data points and their representatives are at least locally minimized, and that representatives and mean squared errors resulting from aggregation to coarser resolutions will be properly preserved. Details of the algorithm like the one used to produce these summaries can be found in Braverman et al. (2003).

Starting with a monthly summary of MISR cloud data at one degree resolution, our challenge is to discover and understand how relationships among grid cell distributions change spatially, and over different resolutions. In other words, instead of examining spatial patterns of average behavior and variability only, we want to examine spatial patterns of other distributional characteristics such as the number of modes, presence of outliers, and nonlinear regressions. This requires interactively comparing summaries of different grid cells, and of aggregated spatial areas. Thus, we want to quickly visualize summaries, and construct summaries of summaries in hierarchical fashion. The main subject of this paper is the Java tool L3View, written to facilitate this.

2 L3View

The basic data structure underlying L3View is a 180×360 array of objects called L3Cell's. An L3Cell contains a variable-length vector of Cluster objects, with the number of objects depending on grid cell data complexity. A Cluster records a three-dimensional cluster representative, a cluster count, and a within-cluster mean squared error. L3View presents a map of the world, and when the user clicks on it the with the mouse, L3View translates the mouse position into geographic coordinates. L3View opens a separate window, and displays a simple, multivariate visualization of the summary for the one degree grid cell at that location. Further, the user can select a subregion of the map with a rubberband box, and choose to summarize summaries of all grid cells within the box. This too is shown in a new window using the simple multivariate display.

2.1 Main Map and Control Panel

The left panel of Figure 2 is a screenshot of the main L3View control panel. L3View uses Java Swing components to interact with users. The image displayed is constructed from information in the grid cells' clusters and combined with a GIF file containing continental outlines using Java image processing functions. L3View knows the position of the mouse in a graphic coordinate space native to the underlying Java object type, JPanel. L3View has methods to convert back and forth between this coordinate system, the 180×360 grid, and latitude and longitude. Latitude and longitude are displayed interactively as the mouse is moved, and the tool knows when the mouse is

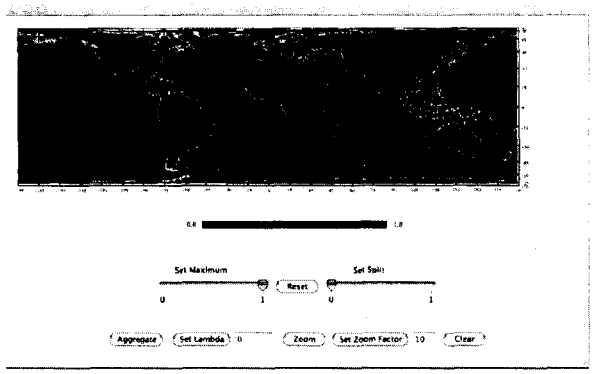

Figure 2: L3View main control panel showing MISR cloud fraction for March 2000.

clicked, dragged, or leaves the map area. Clicking on a grid cell spawns a GraphView window, which contains three graphics for visualizing the clusters representing that grid cell's data.

If the mouse is used to isolate a rectangular geographic region with a rubberband box, L3View calculates the corresponding geographic and index limits. These are subsequently used in two cases. First, if the Zoom button is pushed, a new window containing a magnified image of the isolated area is spawned. Second, if the Aggregate button is pushed, all clusters from all grid cells inside the box are summarized, and the result is displayed in a new GraphView window. The lambda text box accepts user specified values for a parameter of the summarization algorithm that specifies how much data reduction is applied. This is discussed in Section 3.

Finally, the Set Maximum and Set Split sliders are used to study spatial patterns in the cumulative distribution function of the display variable. Set Maximum truncates the upper end of the color scale so that all grid cells with display values at or above the maximum display white. Set split is similar: all values above the split value are displayed white, while all values below the split value display in black.

2.2 The GraphView Window

The GraphView window is a simple, three panel multivariate visualization of a set of clusters. A typical GraphView window is shown in Figure 3. It includes two bar plots and a parallel coordinate plot. The bar plots are two instances of the same class, instantiated to display cluster counts and mean squared errors (distortions). Each has one bar per cluster, and bars are sorted in order of increasing cluster count. Actual values of counts and distortions

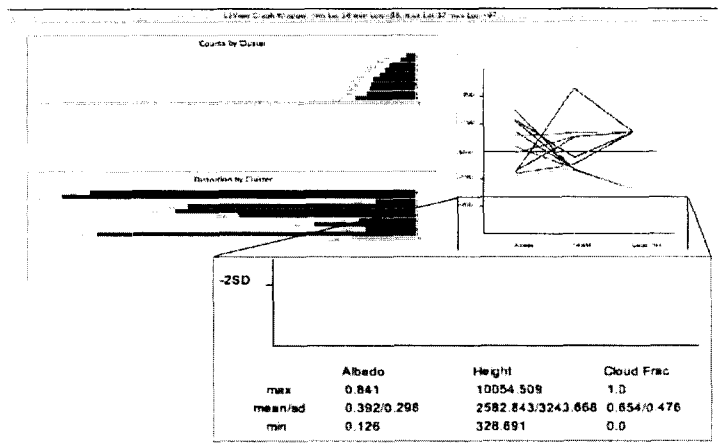

Figure 3: A GraphView window showing the summary of MISR albedo, height and cloud indicator for the grid cell with southwest corner 36°N, 98°W over northern Oklahoma. A zoom-in view of the parallel coordinate plot legend is shown in superimposed box.

relative to the norms of corresponding cluster representatives, are shown at the bars' left edges. Though not apparent in these black and white figures, bars are colored using a scheme that transitions smoothly from blue to red with increasing count.

The parallel coordinate plot occupies the right side of the window. Each line plot shows the representative values of albedo, scene height, and cloudiness for a single cluster on scales normalized using the global means and standard deviations. These are shown at the bottom of the parallel coordinate plot area. Lines are color coded to match bars in the other two panels so users can see which representatives belong to which clusters. In addition, clicking on any bar or any line highlights the bars and line in all plot corresponding to that cluster.

GraphView windows are spawned to visualize a set of clusters, either for a single grid cell or when a set of grid cells are to be summarized collectively. In the latter case, one could simply display the entire collection of clusters, but that would become more unwieldy for large areas as more clusters are included. Complexity of the parallel coordinate plots could grow to the point where it is impossible to resolve individual lines. Therefore, distributions represented by cluster sets must be summarized before they are displayed. The next section describes the theoretical rationale for this.

3 Hierarchical Aggregation and Quantization

Braverman (2002) described how entropy–constrained vector quantization (ECVQ; Chou, Lookabaugh and Gray, 1989) is modified to function as a data reduction tool for large, spatial data sets. The basic idea is to partition these data into one degree spatial subsets, and use ECVQ to cluster the subsets in a coordinated way. ECVQ is a randomized, iterative algorithm similar to K-means, except it minimizes the expected value of the penalized loss function,

$$L_\lambda(\boldsymbol{X}, \alpha(\boldsymbol{X})) = \|\boldsymbol{X} - q(\boldsymbol{X})\|^2 + \lambda \left[-\log \frac{N_{\alpha(\boldsymbol{X})}}{N} \right]. \quad (1)$$

$\boldsymbol{X}$ represents a randomly drawn observation from the empirical distribution of the grid cell's data. $\alpha(\boldsymbol{X})$ is an integer that specifies the id number of the cluster to which $\boldsymbol{X}$ is assigned, and $q(\boldsymbol{X})$ is the corresponding cluster centroid. N is the total number of data points in the grid cell, $N_{\alpha(\boldsymbol{X})}$ is the number of data points assigned to the same cluster as $\boldsymbol{X}$, and the logarithm is base two. λ is a fixed parameter that specifies how important the second term on the right in Equation (1) is. For K-means, one must specify K, the number of clusters a priori. For ECVQ, one must specify K, the maximum allowable number of clusters, and λ. The algorithm then determines the number of clusters and the assignment of data points to them. We added a final step in which each data point is subsequently reassigned to the cluster with the nearest euclidian distance representative, and the representatives updated again. This ensures cluster representatives are centroids of cluster members, and mean squared errors between data points and their representatives are minimized. In Braverman (2003) we introduced a further modification of ECVQ in which $\boldsymbol{X}$ is a random variable having the distribution of $q(\boldsymbol{X})$ rather than the original empirical distribution of the data. In other words, we allow realizations to have unequal mass. That is precisely the situation in which we find ourselves when summarizing sets of clusters formed by combining multiple grid cells.

Consider Figure 4. It shows a schematic representation of a one degree spatial grid. Each grid cell contains a summary instantiated as an L3Cell object. Figure 4 also shows a two degree grid cell superimposed, and suppose we want to summarize the four, one degree L3Cell's inside. Let $\boldsymbol{X}_{1uv}$ be a random variable having the distribution of the summary for the one degree grid cell with southwest corner at row u and column v. Suppose this grid cell is the lower-left most grid cell in the light box in Figure 4, and denote the other three grid cells' random variables by $\boldsymbol{X}_{1(u+1)v}$, $\boldsymbol{X}_{1u(v+1)}$, and $\boldsymbol{X}_{1(u+1)(v+1)}$. At coarser, two degree resolution the light box is represented by $\boldsymbol{X}_{2uv}$,

$$\boldsymbol{X}_{2uv} = \sum_{i=0}^{1} \sum_{j=0}^{1} \boldsymbol{X}_{1(u+i)(v+j)} 1[V = v_{(u+i)(v+j)}],$$

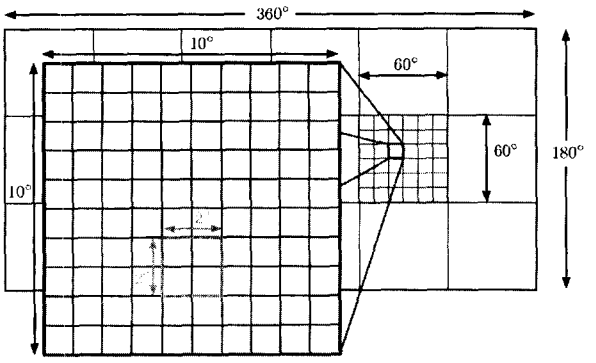

Figure 4: Schematic representation of a gridded map. The large rectangle represents a 180 × 360 array shown broken into 3 × 6 = 18, 60 × 60 arrays. Each of these is further subdivided into a 6 × 6 array. Each cell in the 6 × 6 array is a 10 × 10 arrangement of one degree grid cells. The lighter box illustrates how four one degree grid cells can make up a grid cell at coarser, two degree resolution.

with

$$P\left(V = v_{(u+i)(v+j)}\right) = \frac{N_{(u+i)(v+j)}}{\sum_{i=0}^{1}\sum_{j=0}^{1} N_{(u+i)(v+j)}},$$

and N_{ij} is the total number of data points represented by the summary of the corresponding grid cell. In other words, $\boldsymbol{X}_{2uv}$ is a mixture of $\boldsymbol{X}_{1uv}$, $\boldsymbol{X}_{1(u+1)v}$, $\boldsymbol{X}_{1u(v+1)}$, and $\boldsymbol{X}_{1(u+1)(v+1)}$ with weights equal to the proportions of the total count represented by $\boldsymbol{X}_{2uv}$ contributed by each one degree cell. The idea is illustrated on the left side of Figure 5, which shows the mixture distribution positioned directly above the four component distributions. Any nesting of fine-scale grid cells in a coarser grid can be represented in a similar way, and ensures mass, expectation, and mean squared error are all properly preserved between resolutions.

If data reduction were not a concern, we could proceed directly to visualizing mixture distributions like the middle layer in Figure 5. However, the greater the number of grid cells being aggregated, the greater the number of support points in the mixture, and the number of corresponding clusters. So, we compress the mixture distribution using a mass-weighted version of ECVQ described in Braverman et. al., (2003), but implemented here in Java with the user specifying λ directly via the "Set Lambda" button and text box in the main control panel. K, the maximum number of clusters is nominally set to 10, and the default value of λ is zero, thus essentially implementing the K-means. If λ is changed to a positive value, the algorithm becomes ECVQ.

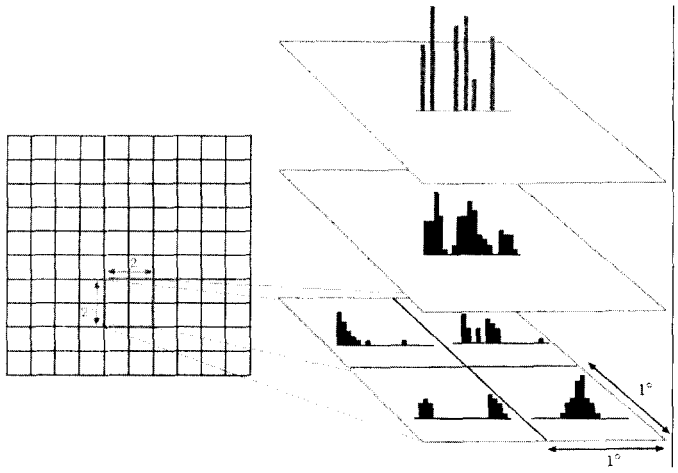

Figure 5: A hierarchy of distributions within a two degree spatial region. The bottom square on the right corresponds to the $2° \times 2°$ area, and shows conceptual representations of cluster sets for constituent one degree grid cells as histograms. The middle layer on the right depicts the mixture distribution formed by the union of the cluster sets from the one degree cells. The top layer is the reduced distribution after summarization.

By first considering aggregated distributions for large areas, and then systematically summarizing subregions, we can begin to understand how the prevalence of various types of phenomena change spatially. The next section demonstrates how this can be done.

4 Visual Data Mining

As an example of how a scientist might use L3View for data exploration, we focus on an area in central Africa shown in Figure 6. The rectangular region extends from latitude 1°S to latitude 9°N, and from longitude 11°E to 31°E. The background L3View image shows cloud fraction. There is a clear difference between the northern and southern parts of this region, approximately demarcated by the horizontal dashed line in embedded, zoomed-in view. The southern area is very cloudy, and the northern area contains grid cells varying cloudiness. This is consistent with the climatological location of a persistent band of clouds called the Inter-Tropical Convergence Zone (ITCZ). The lower panel of Figure 6 shows the GraphView window of the summary of the entire 10×20 degree region.

The region contains 200 grid cells, with a total of 2,099 clusters. These

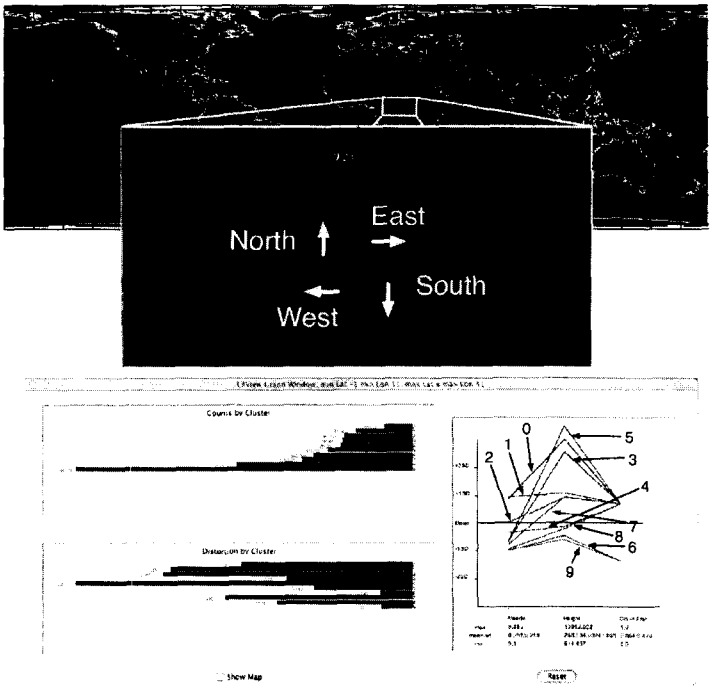

Figure 6: Screenshots from L3View visualization of central Africa. Top: Main L3View map with embedded zoom of central Africa. Bottom: GraphView window for the entire, aggregated central area. All the parallel coordinate plots are annotated to show the cluster id numbers corresponding to the individual line plots.

represent 6,205,769 original MISR albedo–height–cloud indicator vectors. We begin by aggregating the whole region using the default value of $\lambda = 0$ and the number of clusters, K, set to 15. The resulting GraphView summary is shown in the lower panel of Figure 6. The figure is small making the graph labels difficult to see, but we can see from the bar chart of cluster counts that one cluster dominates in size. Using L3View interactively, we find that this is cluster 9, and it contains about 30 percent of the distribution's mass. Cluster 9 corresponds to one of two clear clusters, 6 being the other. Cluster 6 accounts for another eight percent of the distribution's mass. 9 and 6 have representatives with low albedo, low height, and are clear cloud indicators. This is a dark, vegetated region of jungle. Areas to its north show significant numbers of low altitude, bright, clear scenes. This is the Sahara desert.

The remaining clusters have cloudy representatives, and form three subgroups. Clusters 8 and 4 constitute a subgroup with low albedo and below average height. Clusters 1, 2, and 7 form a second subgroup. Their heights are nearly one standard deviation above the mean, but their albedos range from nearly one standard deviation below to one standard deviation above average. The final subgroup is characterized by very large heights, two standard deviations above the mean at least. They too show a range of albedos similar to that of the second subgroup. These high clouds are likely the tops of thunderstorms prevalent in central Africa at this time of year, and the surrounding cloud formations. The first two subgroups are more mysterious. Clusters 1, 2, and 7 could be low and mid-level cumulus and stratus clouds. 8 and 4 are possibly dust, clear land surface misclassified as cloud, or simply dark, low clouds, as implied by the classification.

Noting the relatively sharp difference in cloud fractions between the northern and southern areas, we separately summarize them are shown in Figure 7. Signatures of southern region representatives look much like signatures for the region as a whole. The north's representatives also look roughly like those of the whole region except clusters similar to 0 and 4 are missing. The absence of clusters similar to cluster 0 in the lower panel is encouraging, since this cluster represents deep convective clouds. Corroborating sources indicate these are in fact absent in this region at this time.

These distributional differences are summarized in Table 1. Not surprisingly the joint distribution shows that the south is cloudier than the north. The fact that the south is dominated by low clouds while the north is dominated by mid-level clouds is less obvious but clear from the conditional distribution.

	Joint			Conditional	
Type	*North*	*South*	*Total*	*North*	*South*
Clear	0.330	0.064	0.394		
Low cloud	0.060	0.171	0.231	0.273	0.442
Mid-level cloud	0.094	0.114	0.207	0.426	0.295
High cloud	0.066	0.101	0.168	0.301	0.263
Total cloudy	0.220	0.386	0.606	1.000	1.000
Total	0.550	0.450	1.000		

Table 1: Joint and conditional distributions of cloud type/clear and location. Columns 2, 3, and 4 show the full, joint distribution. Columns 5 and 6 show the conditional distribution of cloud type given cloudy scene, and location.

The presence of low, dark clouds in both the north and the south at this time of year is something of a surprise. To see if these clouds can be attributed to specific areas, we subdivided the north and south regions into east and west. We found no distributional differences related to east-west

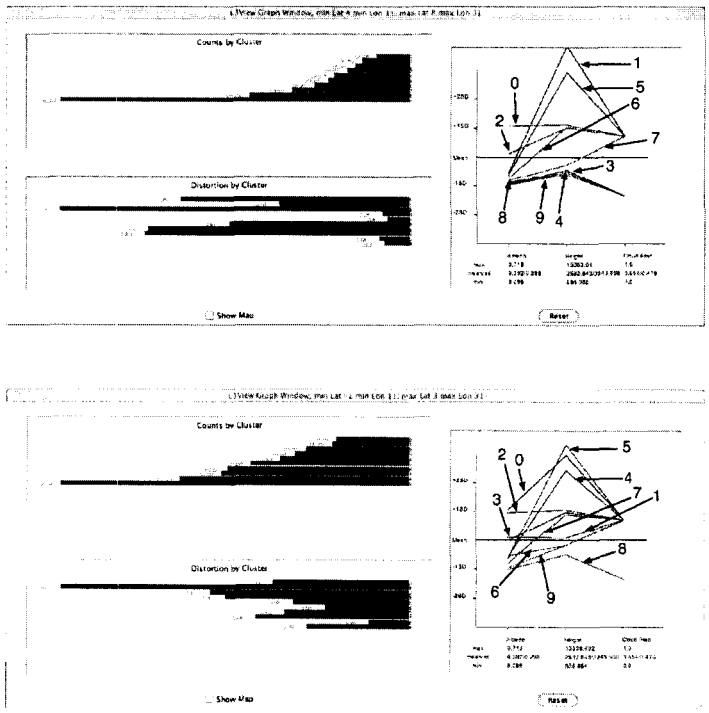

Figure 7: Top: GraphView window for the aggregated area in the northern half of the region (above the dashed line). Bottom: GraphView window for the aggregated area in the southern half of the region (below the dashed line).

division for either the north or south. We then investigated areas along the prominent clear/cloudy boundary in Figure 6, and contrasted them to areas away from the boundary. We did this separately for east and west, but none of these visualizations revealed definitive distributional differences. We are therefore reasonably confident that Table 1 tells a complete story.

5 Discussion

The example of the previous section is a small scale, simple example of one way we think L3View may be useful for exploring spatial summaries of satellite data. Guided by the background map in the main L3View window, we focused on an area of interest, and hierarchically examined the data distribution summaries. We discovered that, in addition to the cloud fraction

differences apparent from the background image, there is also a difference in the types of cloud present in the northern and southern parts of the region. We will have to perform many more exercises like this one to gain confidence that summarized data have enough detail to be scientifically useful, and to gain experience interpreting physically what we see in L3View.

Two main computational issues are brought to light in this exercise. First, L3View's implementation of ECVQ/K-means is not fast enough to summarize large geographic regions in reasonable time. One would ideally like to summarize whole hemispheres in the same sort of hierarchical exploration performed here. Second, we have not yet made use of the tool's ability to summarize the same data for different values of λ's. We would like to look at data at different quantization resolutions as well as different spatial ones. We believe there is important information in how distributions collapse as greater data reduction is imposed. To achieve greater interactivity both these issues must be addressed. We look forward to working on these and other improvements to L3View as it matures. We also eagerly anticipate working with our geoscience colleagues to better understand the connection between global physical processes and their expression through rich, Earth Observing System data sets.

References

[1] Braverman, Amy, Fetzer, Eric, Eldering, Annmarie, Nittel, Silvia, and Leung, Kelvin (2003), "Semi-streaming Quantization for Remote Sensing Data", *Journal of Computational and Graphical Statistics*, **12**, 4, pp. 759–780.

[2] Braverman, Amy (2002), "Compressing Massive Geophysical Datasets Using Vector Quantization", *Journal of Computational and Graphical Statistics*, **11**, 1, pp. 44-62.

[3] Chou, P.A., Lookabaugh, T., and Gray, R.M. (1989), "Entropy-constrained Vector Quantization," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, **37**, pp. 31-42.

Acknowledgement: The authors would like to thank Eric Fetzer, Annmarie Eldering and Barbara Gaitley for their helpful comments. This work was performed at the Jet Propulsion Laboratory, California Institute of Technology, under contract with the National Aeronautics and Space Administration.

Address: Amy Braverman is a statistician and Scientist at the Jet Propulsion Laboratory, California Institute of Technology, Mail Stop 169-237, 4800 Oak Grove Drive, Pasadena, CA 91109-8099. Brian Kahn is a graduate student in the Department of Atompsheric Science, UCLA, 405 Hilgard Avenue, Los Angeles, CA 90095.

E-mail: `Amy.Braverman@jpl.nasa.gov, kahn@atmos.ucla.edu.`